WE CLAIM:

1. A method for enabling location independent and location transparent interaction between a program and a user, the program having been launched at a first location and having a program state data structure for storing at least the program state, the method comprising the steps of:

initiating a program status request by the user;

determining the current location of the program;

checking the program state to ascertain program status; and

interacting with said program based upon said program status.

2. The method of Claim 1 wherein said interacting with said program comprises:

retrieving, from the program, output contents to display to the user; and

displaying the output contents to said user.

3. The method of Claim 1 wherein said interacting with said program comprises:

requesting input variables from said user;

sending any received input values to the current location; and

incorporating the received input values into said program state data structure.

4. The method of Claim 1 wherein the program is a mobile agent.

5. The method of Claim 1 wherein the program is a mobile script.

6. The method of Claim 1 where the user is a mobile user.

7. The method of Claim 2 further comprising the step of maintaining an output buffer and wherein said retrieving comprises the step of retrieving the output contents from said output buffer.

8. The method of Claim 1 wherein the initiating step comprises the steps of:

initiating the status request at a client machine; and

forwarding the status request to the first location at which said program was launched.

9. The method of Claim 8 wherein said program comprises a mobile program which executes a portion of its code at each of a plurality of execution servers and wherein the determining step comprises the steps of:

transmitting the status request to each execution server at which the program has executed a portion of its code; and

determining, at each execution server, whether the program is currently running locally.

10. The method of Claim 9 wherein each of said plurality of execution servers maintains routing information for said program

~~and wherein said determining further comprises the step, if said program is not currently running locally, of consulting said routing information to ascertain at least one successive execution server to which the program has been routed.~~

11. A method for enabling a user to provide input values to a running program before the program needs the input values, comprising the steps of:

maintaining a bag buffer of variable/value pairs in the program;

receiving a communication, including input values, from the user; and

temporarily storing said input values in said bag buffer.

12. The method of Claim 11 wherein said program subsequently searches through contents of the bag buffer to locate needed input values before requesting input from said user.

Sub A2

~~13. The method of Claim 2 further comprising the step of maintaining a bag buffer in the program and wherein the retrieving step comprises the steps of:~~

~~searching, in the bag buffer, for input values associated with the input variables;~~

~~updating, if found, the input variables with the input values;~~

~~disposing, in an input buffer, the input variables, if not found; and~~

optionally notifying the user via electronic means if no suitable values are found in the bag buffer.

14. The method of Claim 13 wherein the electronic means is a pager.

15. The method of Claim 13 wherein the electronic means is a beeper.

16. The method of Claim 13 wherein the electronic means is electronic mail.

17. The method of Claim 13 wherein the electronic means is a smart telephone.

18. A computer program data structure comprising;

an output buffer for storing output values to be displayed to a user;

an input buffer for storing values for which user input of variables is required; and

a program state buffer for storing at least the present state of said program.

19. The data structure of Claim 18 further comprising a bag buffer for storing input variables.

20. The data structure of Claim 19 wherein the bag buffer is a array data structure.

21. The data structure of Claim 19 wherein the bag buffer is a hash table data structure.

22. The data structure of Claim 19 wherein the bag buffer is a tuple space data structure.

23. An execution shell for a mobile program comprising:

a routing component for maintaining routing information regarding said mobile program;

a processor component for processing user status requests related to said program; and

an execution component for executing at least part of said program.

24. The execution shell of Claim 23 further comprising a data handling component for receiving user input and storing same in at least one data structure for said program.